Notice of Allowability	Application No.	Applicant(s)	
	09/764,962	DOUCEUR ET AL.	
	Examiner	Art Unit	
	Peter Poltorak	2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *11/29/05*.

2. ☒ The allowed claim(s) is/are *19-26,29,30,32-39,47,49,50,52-54,60-64,69-73 and 75-81*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted:
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date 9/23/03, 8/05/04, 10/14/04, 10/05/05 ✓
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance } together
9. ☐ Other _____.

DETAILED ACTION

1. This Office Action is in response to Applicant's amendment filed on 11/29/05.

Examiner Amendment

2. An Examiner's Amendment to the record appears below. Should the changes and/or additions be unacceptable to Applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the Issue Fee.

   The following changes were authorized (and permission to make same by Authorization for this Examiner's Amendment was given in a telephone interview with Allan Sponseller (509 324-9256) on February 14, 2006.

Replace the current claims with the following claims:

--

1-18. (Canceled).

19. (Currently amended) A computer-implemented method comprising:

receiving an identifier;

generating, based on the identifier, a mapped identifier;

encoding the mapped identifier, wherein encoding the mapped identifier comprises:

reversing an order of characters in the mapped identifier;

removing, from the reversed mapped identifier, all trailing characters of a particular type;

initializing ~~the encoded~~ an identifier string with a string of one bits equal in number to a number of trailing characters removed from the reversed mapped identifier followed by a zero bit;

selecting a first character from the reversed mapped identifier;

encoding the first character using a first coding table;

modifying the identifier string by adding, to the ~~encoded~~ identifier string, a series of zero bits followed by the encoded first character;

for each additional character in the reversed mapped identifier,

selecting a next character in the reversed mapped identifier,

encoding the next character using a second coding table,

modifying the previously modified identifier string by adding, to the ~~encoded~~ previously modified identifier string, a series of zero bits followed by the encoded next character; and

removing any trailing zero bits and the one bit preceding the trailing zero bits from the ~~encoded~~ identifier string, wherein upon completing removing any trailing zero bits and the one bit the identifier string is the encoded mapped identifier; and

encrypting the encoded mapped identifier.

20. (Original) A method as recited in claim 19, wherein the identifier comprises one of: a file name, a folder name, and a directory name.

21. (Original) A method as recited in claim 19, further comprising:

generating, based on the mapped identifier, a decasified identifier and corresponding case information;

wherein the encoding comprises encoding the decasified identifier; and

wherein the encrypting comprises encrypting both the encoded decasified identifier and the case information.

22. (Original) A method as recited in claim 21, wherein generating the decasified identifier and corresponding case information comprises:

for each character that has both an upper-case and a lower-case form, storing the character in upper-case form and recording in the case information whether the character was in upper-case form or lower-case form.

23. (Previously presented) A method as recited in claim 22, further comprising:

storing the character in upper-case form only if the character is one of a particular set of characters; and

storing the character without altering its case if the character is not one of the particular set of characters.

24. (Original) A method as recited in claim 23, wherein the particular set of characters comprises the extended ASCII character set.

25. (Original) A method as recited in claim 19, wherein the generating comprises generating the mapped identifier only if the received identifier is syntactically legal.

26. (Original) A method as recited in claim 19, wherein the encoding comprises encoding the mapped identifier only if the received identifier is syntactically legal.

27. (Canceled).

28. (Canceled).

29. (Original) A method as recited in claim 19, wherein generating the mapped identifier comprises:

checking whether the identifier is equal to one of a plurality of illegal identifiers;

if the identifier is not equal to one of the plurality of illegal identifiers, then checking whether the identifier is equal to one of the plurality of illegal identifiers followed by one or more particular characters;

if the identifier is not equal to one of the plurality of illegal identifiers followed by one or more particular characters, then using the identifier as the mapped identifier; and

if the identifier is equal to one of the plurality of illegal identifiers followed by one or more particular characters, then using as the mapped identifier the identifier with one of the particular characters removed.

30. (Original) A method as recited in claim 29, wherein the particular character comprises an underscore.

31. (Canceled).

32. (Previously presented) A method as recited in claim 19, wherein the characters of a particular type are the characters that are coded to zero using the first coding table.

33. (Previously presented) A method as recited in claim 19, wherein the first coding table and the second coding table are Huffman coding tables.

34. (Previously presented) A method as recited in claim 19, wherein each coding in the first coding table is the same as a corresponding coding in the second coding table, but the second coding table codes additional characters not coded by the first coding table.

35. (Currently amended) A method as recited in claim 19, wherein for the ~~first character and~~ each additional character in the reversed mapped identifier, encoding the character only if a set of leading bits of the character are zero, and further comprising modifying the previously modified identifier string by adding the character to the ~~encoded~~ previously modified identifier string if the set of leading bits of the character are not zero.

36. (Currently amended) A computer-implemented method comprising:

receiving an identifier;

generating, based on the identifier, a mapped identifier;

encoding the mapped identifier, wherein encoding the mapped identifier comprises:

reversing an order of characters in the mapped identifier;

removing, from the reversed mapped identifier, all trailing characters of a particular type;

initializing ~~the encoded~~ an identifier string with a string of one bits equal in number to a number of trailing characters removed from the reversed mapped identifier followed by a zero bit;

selecting a first character from the reversed mapped identifier;

encoding the first character using a first coding table;

modifying the identifier string by adding, to the ~~encoded~~ identifier string, a series of zero bits followed by the encoded first character;

for each additional character in the reversed mapped identifier,

selecting a next character in the reversed mapped identifier,

encoding the next character using one of a plurality of additional coding tables,

modifying the previously modified identifier string by adding, to the ~~encoded~~ previously modified identifier string, a series of zero bits followed by the encoded next character; and

removing any trailing zero bits and the one bit preceding the trailing zero bits from the ~~encoded~~ identifier string, wherein upon completing removing any trailing zero bits and the one bit the identifier string is the encoded mapped identifier; and

encrypting the encoded mapped identifier.

37. (Previously presented) A method as recited in claim 19, wherein encrypting the encoded mapped identifier comprises using a block cipher to encrypt the encoded identifier.

38. (Previously presented) A method as recited in claim 19, wherein encrypting the encoded mapped identifier comprises using cipher block chaining to encrypt the encoded identifier.

39. (Previously presented) A method as recited in claim 19, wherein the encrypting comprises encrypting the encoded mapped identifier to generate, using a block cipher, a ciphertext having a fixed size.

40-46 (Canceled).

47. (Currently amended) One or more computer-readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a computer, causes the one or more processors to perform acts including:

receiving a plaintext identifier;

generating a ciphertext by encrypting the plaintext identifier only if the plaintext identifier is syntactically legal, wherein generating the ciphertext comprises:

generating, based on the plaintext identifier, a mapped identifier;

encoding the mapped identifier, wherein encoding the mapped identifier comprises:

reversing an order of characters in the mapped identifier;

removing, from the reversed mapped identifier, all trailing characters of a particular type;

initializing ~~the encoded~~ an identifier string with a string of one bits equal in number to a number of trailing characters removed from the reversed mapped identifier followed by a zero bit;

selecting a first character from the reversed mapped identifier;

encoding the first character using a first coding table;

modifying the identifier string by adding, to the ~~encoded~~ identifier string, a series of zero bits followed by the encoded first character;

for each additional character in the reversed mapped identifier,

selecting a next character in the reversed mapped identifier,

encoding the next character using a second coding table,

modifying the previously modified identifier string by adding, to the ~~encoded~~ previously modified identifier string, a series of zero bits followed by the encoded next character; and

removing any trailing zero bits and the one bit preceding the trailing zero bits from the ~~encoded~~ identifier string, wherein upon completing removing any trailing zero bits and the one bit the identifier string is the encoded mapped identifier; and

encrypting the encoded mapped identifier; and

wherein the encrypting allows another device to verify, without decrypting the ciphertext, that the plaintext identifier is not identical to another plaintext identifier maintained by the other device.

48. (Canceled).

49. (Previously presented) One or more computer-readable media as recited in claim 47, wherein generating the ciphertext further comprises:

generating, based on the mapped identifier, a decasified identifier and corresponding case information;

wherein the encoding comprises encoding the decasified identifier; and

wherein the encrypting comprises encrypting both the encoded decasified identifier and the case information.

50. (Previously presented) One or more computer-readable media as recited in claim 47, wherein generating the mapped identifier comprises:

checking whether the plaintext identifier is equal to one of a plurality of illegal identifiers;

if the plaintext identifier is not equal to one of the plurality of illegal identifiers, then checking whether the plaintext identifier is equal to one of the plurality of illegal identifiers followed by one or more particular characters;

if the plaintext identifier is not equal to one of the plurality of illegal identifiers followed by one or more particular characters, then using the plaintext identifier as the mapped identifier; and

if the plaintext identifier is equal to one of the plurality of illegal identifiers followed by one or more particular characters, then using as the mapped identifier the plaintext identifier with one of the particular characters removed.

51. (Canceled).

52. (Currently amended) One or more computer-readable media as recited in claim 47, wherein each coding in the first coding table is the same as a corresponding coding in the second coding table, but the second coding table codes additional characters not coded by the first coding table.

53. (Currently amended) One or more computer-readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a computer, causes the one or more processors to perform acts including:

receiving a plaintext identifier;

generating a ciphertext by encrypting the plaintext identifier only if the plaintext identifier is syntactically legal, wherein generating the ciphertext comprises:

generating, based on the plaintext identifier, a mapped identifier;

encoding the mapped identifier, wherein encoding the mapped identifier comprises:

reversing an order of characters in the mapped identifier;

removing, from the reversed mapped identifier, all trailing characters of a particular type;

initializing ~~the encoded~~ an identifier string with a string of one bits equal in number to a number of trailing characters removed from the reversed mapped identifier followed by a zero bit;

selecting a first character from the reversed mapped identifier;

encoding the first character using a first coding table;

modifying the identifier string by adding, to the ~~encoded~~ identifier string, a series of zero bits followed by the encoded first character;

for each additional character in the reversed mapped identifier,

selecting a next character in the reversed mapped identifier,

encoding the next character using one of a plurality of additional coding tables,

modifying the previously modified identifier string by adding, to the ~~encoded~~ previously modified identifier string, a series of zero bits followed by the encoded next character; and

removing any trailing zero bits and the one bit preceding the trailing zero bits from the ~~encoded~~ identifier string, wherein upon completing removing any trailing zero bits and the one bit the identifier string is the encoded mapped identifier; and

encrypting the encoded mapped identifier; and

wherein the encrypting allows another device to verify, without decrypting the ciphertext, that the plaintext identifier is not identical to another plaintext identifier maintained by the other device.

54. (Previously presented) One or more computer-readable media as recited in claim 47, wherein encrypting the encoded mapped identifier comprises using a block cipher to encrypted the encoded identifier.

55-59 (Canceled).

60. (Currently amended) One or more computer-readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a computer, causes the one or more processors to perform acts including:

receiving a plaintext directory entry;

verifying that the plaintext directory entry is syntactically legal;

encrypting the plaintext directory entry only if the plaintext directory entry is syntactically legal, wherein encrypting the plaintext directory entry comprises:

generating, based on the plaintext directory entry, a mapped identifier;

encoding the mapped identifier, wherein encoding the mapped identifier comprises:

reversing an order of characters in the mapped identifier;

removing, from the reversed mapped identifier, all trailing characters of a particular type;

initializing ~~the encoded~~ an identifier string with a string of one bits equal in number to a number of trailing characters removed from the reversed mapped identifier followed by a zero bit;

selecting a first character from the reversed mapped identifier;

encoding the first character using a first coding table;

modifying the identifier string by adding, to the ~~encoded~~ identifier string, a series of zero bits followed by the encoded first character;

for each additional character in the reversed mapped identifier,

selecting a next character in the reversed mapped identifier,

encoding the next character using a second coding table,

modifying the previously modified identifier string by adding, to the ~~encoded~~ previously modified identifier string, a series of zero bits followed by the encoded next character; and

removing any trailing zero bits and the one bit preceding the trailing zero bits from the ~~encoded~~ identifier string; wherein upon completing removing any trailing zero bits and the one bit the identifier string is the encoded mapped identifier; and

encrypting the encoded mapped identifier;

communicating the encrypted directory entry to another device; and

wherein the encrypting allows the other device to verify, without decrypting the encrypted directory entry, that the directory entry is not identical to any other directory entry maintained by the other device.

61. (Original) One or more computer-readable media as recited in claim 60, wherein the computer is part of a serverless distributed file system.

62. (Original) One or more computer-readable media as recited in claim 60, wherein the plaintext directory entry comprises a file name.

63. (Original) One or more computer-readable media as recited in claim 60, wherein the plaintext directory entry comprises a directory name.

64. (Original) One or more computer-readable media as recited in claim 60, wherein the plaintext directory entry comprises a folder name.

65-68. (Canceled).

69. (Previously presented) One or more computer-readable media as recited in claim 60, wherein encrypting the plaintext directory entry further comprises:

generating, based on the mapped identifier, a decasified identifier and corresponding case information;

wherein the encoding comprises encoding the decasified identifier; and

wherein the encrypting comprises encrypting both the encoded decasified identifier and the case information.

70. (Previously presented) One or more computer-readable media as recited in claim 60, wherein generating the mapped identifier comprises generating the mapped identifier only if the received plaintext directory entry is syntactically legal.

71. (Previously presented) One or more computer-readable media as recited in claim 60, wherein the encoding comprises encoding the mapped identifier only if the received plaintext directory entry is syntactically legal.

72. (Previously presented) One or more computer-readable media as recited in claim 60, wherein generating the mapped identifier comprises:

checking whether the plaintext directory entry is equal to one of a plurality of illegal identifiers;

if the plaintext directory entry is not equal to one of the plurality of illegal identifiers, then checking whether the plaintext directory entry is equal to one of the plurality of illegal identifiers followed by one or more particular characters;

if the plaintext directory entry is not equal to one of the plurality of illegal identifiers followed by one or more particular characters, then using the plaintext directory entry as the mapped identifier; and

if the plaintext directory entry is equal to one of the plurality of illegal identifiers followed by one or more particular characters, then using as the mapped identifier the plaintext directory entry with one of the particular characters removed.

73. (Original) One or more computer-readable media as recited in claim 72, wherein the particular character comprises an underscore.

74. (Canceled).

75. (Previously presented) One or more computer-readable media as recited in claim 60, wherein each coding in the first coding table is the same as a corresponding coding in the second coding table, but the second coding table codes additional characters not coded by the first coding table.

76. (Previously presented) One or more computer-readable media as recited in claim 60, wherein the characters of a particular type are the characters that are coded to zero using the first coding table.

77. (Previously presented) One or more computer-readable media as recited in claim 60, wherein the first coding table and the second coding table are Huffman coding tables.

78. (Previously presented) One or more computer-readable media as recited in claim 60, wherein each coding in the first coding table is the same as a corresponding coding in the second coding table, but the second coding table codes additional characters not coded by the first coding table.

79. (Previously presented) One or more computer-readable media as recited in claim 60, wherein for ~~the first character and~~ each additional character in the reversed mapped identifier, encoding the character only if a set of leading bits of the character

are zero, and further comprising modifying the previously modified identifier string by adding the character to the ~~encoded~~ previously modified identifier string if the set of leading bits of the character are not zero.

80. (Previously presented) One or more computer-readable media as recited in claim 60, wherein encrypting the encoded mapped identifier comprises using a block cipher to encrypt the encoded identifier.

81. (Original) One or more computer-readable media as recited in claim 60, wherein the encrypting further comprises generating, using a block cipher, the encrypted directory entry having a fixed size.

82-87 (Canceled).

--

Allowable Subject Matter

3. Claims 19-26, 29-30, 32-39, 47, 49-50, 52-54, 60-64, 69-73 and 75-81 are allowed.
4. Claims 1-18, 27-28, 31, 40-46, 48, 51, 55-59, 65-68, 74 and 82-87 are canceled.
5. The following is a statement of reasons for the indication of allowable subject matter.
6. Applicant's invention relates to methods and systems for encryption that excludes syntactically illegal plaintext from being encrypted and that enables a party without access to encryption keys to exclude more than one item of ciphertext that decrypts to the same plaintext.

7. The claim language recite the limitation of receiving an identifier, generating, based on the identifier, a mapped identifier, encoding the mapped identifier and encrypting the encoded mapped identifier.

The independent claims 19, 36, 47, 53 and 60 identify the uniquely distinct features of encoding the mapped identifier:

reversing an order of characters in the mapped identifier;

removing, from the reversed mapped identifier, all trailing characters of a particular type;

initializing an identifier string with a string of one bits equal in number to a number of trailing characters removed from the reversed mapped identifier followed by a zero bit;

selecting a first character from the reversed mapped identifier;

encoding the first character using a first coding table;

modifying the identifier string by adding, to the identifier string, a series of zero bits followed by the encoded first character;

for each additional character in the reversed mapped identifier,

selecting a next character in the reversed mapped identifier,

encoding the next character using a second coding table,

modifying the previously modified identifier string by adding, to the previously modified identifier string, a series of zero bits followed by the encoded next character; and

removing any trailing zero bits and the one bit preceding the trailing zero bits from the identifier string, wherein upon completing removing any trailing zero bits and the one bit the identifier string is the encoded mapped identifier.

8. The closest prior art the Unix operating system teaches receiving an identifier, generating, based on the identifier, a mapped identifier, encoding the mapped identifier and Olkin et al. (U.S. Pub. 20030046533) teach encrypting the encoded mapped identifier.
9. However, neither Olkin et al. nor Unix disclose the unique required features (recited above) of encoding the mapped identifier.
10. The prior art, fails to anticipate or fairly suggest the limitation of applicant's independent claims (interpreted in light of the specification), in such a manner that a rejection under 35 U.S.C. 102 or 103 would be proper. As a result the claimed invention is considered to be in condition for allowance as being novel and non-obvious over prior art.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on statement of Reasons for Allowance".

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Peter Poltorak whose telephone number is (571) 272-

3840. The examiner can normally be reached from Monday through Thursday from 9:00 until 5:00, and every other Friday from 9:00 until 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Y. Vu can be reached on (571) 272-3859. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (571) 272-1600.

2/14/6

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100